Interview question on constructor

What is super keywords in Java?

The super keyword **refers to superclass (parent) objects**. It is used to call superclass methods, and to access the superclass constructor. The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name

### 35) What are the differences between the constructors and methods?

|  |  |
| --- | --- |
| **Java Constructor** | **Java Method** |
| A constructor is used to initialize the state of an object. | A method is used to expose the behavior of an object. |
| A constructor must not have a return type. | A method must have a return type. |
| The constructor is invoked implicitly. | The method is invoked explicitly. |
| The Java compiler provides a default constructor if you don't have any constructor in a class. | The method is not provided by the compiler in any case. |
| The constructor name must be same as the class name. | The method name may or may not be same as class name. |

There are many differences between constructors and methods. They are given below.

1. Q. What is constructor?: In [Java](https://www.javatpoint.com/java-tutorial), a constructor is a block of codes similar to the method. It is called when an instance of the [class](https://www.javatpoint.com/object-and-class-in-java) is created. At the time of calling constructor, memory for the object is allocated in the memory It is a special type of method which is used to initialize the object.Every time an object is created using the new() keyword, at least one constructor is called.It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default. A Java constructor cannot be abstract, static, final, and synchronized
2. Q .Types of Constructor?

* **Default Constructor:** default constructor is the one which does not accept any value. The default constructor is mainly used to initialize the instance variable with the default values. It can also be used for performing some useful task on object creation. A default constructor is invoked implicitly by the compiler if there is no constructor defined in the class. A constructor is called "Default Constructor" when it doesn't have any parameter. The purpose of the default constructor is to assign the default value to the objects.
* **Parameterized Constructor:** The parameterized constructor is the one which can initialize the instance variables with the given values. In other words, we can say that the constructors which can accept the arguments are called parameterized constructors. A constructor which has a specific number of parameters is called a parameterized constructor The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.

### Q.What is the purpose of a default constructor?

The purpose of the default constructor is to assign the default value to the objects. The java compiler creates a default constructor implicitly if there is no constructor in the class.

### Q. Can we overload the constructors?

Yes, the constructors can be overloaded by changing the number of arguments accepted by the constructor or by changing the data type of the parameters

Constructor [overloading in Java](https://www.javatpoint.com/method-overloading-in-java) is a technique of having more than one constructor with different parameter lists. They are arranged in a way that each constructor performs a different task. They are differentiated by the compiler by the number of parameters in the list and their types

Q.Is constructor is static or nonstatic?

One of the important property of java constructor is that it can not be [static](https://www.geeksforgeeks.org/static-keyword-java/). We know static keyword belongs to a class rather than the object of a class. A constructor is called when an object of a class is created, so no use of the static constructor. Another thing is that if we will declare static constructor then we can not access/call the constructor from a subclass. Because we know static is allowed within a class but not by a subclass..

### Q) Does constructor return any value?

Yes, it is the current class instance (You cannot use return type yet it returns a value).

Q.Can we declare constructor as private?

**We can declare a constructor private by using the private access specifier**. Note that if a constructor is declared private, we are not able to create an object of the class.

* A **private constructor** does not allow a class to be subclassed.
* A **private constructor** does not allow to create an object outside the class.
* If all the constant methods are there in our class we can use a **private constructor.**
* If all the methods are **static**then we can use a **private constructor.**
* If we try to **extend a class**which is having private constructor **compile time error will occur**.
* **public** **class** PrivateConstructorExample {
* **public** **static** **void** main(String[] args) {
* // **TODO** Auto-generated method stub
* Test.*instanceMethod*();
* }
* }
* **class** Test {
* **private** Test() {
* System.***out***.println("Private constructor called");
* }
* **public** **static** **void** instanceMethod() {
* Test example=**new** Test();
* }
* }
* In the above example, we have created a private constructor of the Test class. Hence, we cannot create an object of the Test class outside of the class.
* This is why we have created a public static method named instanceMethod() inside the class that is used to create an object of the Test class. And from the Main class, we call the method using the class name.

Q.Why do we need constructor private?

If we want security then we have to use private constructor

Don’t want to create objct of class in another class

Q.Can Constructor be a final?

Whenever you make a method final, you cannot **override** it. n other words, constructors cannot be inherited in Java therefore you cannot **override** constructors.

So, writing final before constructors makes no sense. Therefore, java does not allow final keyword before a constructor.

If you try, make a constructor final a compile time error will be generated saying “modifier final not allowed here”.

Q.Can we use constructor in inheritance in Java?

**No, constructors cannot be inherited in Java**. In inheritance sub class inherits the members of a super class except constructors. Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.

Q.Why the constructor cannot abstract in java?

. One of the important property of java constructor is that it can not be [abstract](https://www.geeksforgeeks.org/abstract-keyword-in-java/). If we are declaring a constructor as abstract as we have to implement it in a child class, but we know a constructor is called implicitly when the new keyword is used so it can’t lack a body and also it can not be called as a normal method. Also, if we make a constructor abstract then we have to provide the body later. But we know constructor can not be overridden so providing body is impossible. Hence, what we will do with this abstract constructor when we can not provide implementation to it